

SayTeX: A Math Dictation Tool

Spring 2019 UROP Proposal

Arvid Lunnemark
arvid@mit.edu

Supervisor: Dr. Kyle Keane

February 14, 2019

Project Overview

The aim of this project is to create a product that can be used for dictating LaTeX math formulas reliably, by combining speech-to-text APIs, custom logic, and text-to-LaTeX APIs. This is a continuation of a UROP project I was working on last fall.

This project will be supervised by Dr. Kyle Keane within the Department of Materials Science in the Interactive Materials Education Laboratory. Since this is a software project, the work can and will be done remotely with supervision performed via web conference.

Details

In the fall of 2018, I started working on SayTeX, a product with the goal of converting spoken math formulas into correctly formatted LaTeX code. The current state of the product can be viewed at <https://saytex.xyz> and the current state of the code behind it can be viewed at <https://github.com/arvid220u/saytex>.

SayTeX converts spoken math into LaTeX code in three steps. The first step is transcribing the speech into text, and for this, Microsoft's speech API is used. Currently, only the basic speech API with no modifications is used, and the results can probably be improved by switching to a more specialized API.

After transcribing the speech into text, SayTeX will standardize and sanitize the input. For example, common mistranscriptions such as 'eggs' for 'x' will be dealt with and all supported mathematical operators will be converted into a standardized syntax.

The last step is to convert this standardized syntax into LaTeX code. SayTeX currently has two different ways of doing this: the first is by doing a linear scan that recurses when encountering a set of parentheses. This is fast but not particularly accurate, and each math expression that is supported has had to be manually added by me.

The second way of converting SayTeX syntax into LaTeX code is by using the Wolfram Alpha API, and converting the result from Mathematica code into LaTeX code using a Wolfram Cloud instance. As can be expected, this is a slower but generally more accurate way of converting the math formulas. It should be noted that I eventually aim to replace the Wolfram Alpha API with the custom SayTeX API as soon as it gets sufficiently powerful.

Personal Role & Responsibilities

I will be taking full responsibility for the project, and will work on it independently. I will communicate with Kyle on a regular basis about the progress of the project, and will seek his advice when needed, particularly regarding the use of Wolfram APIs and about integrating it with accessibility software.

Goals

For this semester, I have one big goal for SayTeX: To push it into production to make spoken math a reality for blind students.

To achieve this goal, I have several subgoals:

1. Make it possible for users to edit the given output or mark it as correct, to build up a database of correct mappings from text to LaTeX. This can, in a later stage, be used to help train the model using neural networks or something similar.
2. Update the website to use a modified version of Microsoft's speech API that is more targeted to recognize math words. For this, the aforementioned database will be helpful.
3. Thoroughly investigate the possibility of developing a systematic language that sits in between spoken language and LaTeX. The syntax should be easy and fast to say (e.g., no backslashes and curly braces) but still more precise than natural language.
4. Integrate the product with tools that are used by blind students today.

Personal Statement

My personal motivation for doing this project is two-fold: firstly, I would be able to gain a deeper understanding of the intricacies of LaTeX, Wolfram Alpha and Mathematica, and secondly, I would be given the opportunity to complete a meaningful project on my own, that could actually be useful for other people.